Rexx in the RexxLA Website

2021 International Rexx Symposium
Virtual

Mark Hessling
8 November 2021

Rexx in the RexxLA Website

- Architecture
- Extensions Used
- Rexx Server Pages

Technologies

- RexxServerManager
  - Starts and monitors RexxLA Web Server, rexxla.rexx
- Apache Proxy
  - https://www.rexxla.org redirected to RexxLA Web Server on localhost:8080
- HTML5, CSS; Bootstrap, Javascript for presentation
- MySQL for data storage

Extensions Used

- Rexx/WS
  - Provides http server including SSL
  - Rexx Server Pages
- Rexx/SQL
  - Access to MySQL
    - Now with Substitution variables!
- Rexx/EEC
  - Encryption of passwords
  - HTML Encoding of URLs

Extensions Used (cont.)

- Rexx/JSON
    - Data transmission from Javascript to Rexx and vice versa
- RegUtil (RexxUtil)
    - sysfiletree(), regstemdoover(), etc

Rexx Server Pages

- Embedded Rexx code in html files
    - Dynamic RSP - .rsp
        - Content changes (possibly) each time page is visited
        - *Example* Symposium page
    - Static RSP - .srsp
        - content does not change each time page is visited
        - Generated, cached and served as .html
        - Re-generated when source .srsp or included files changed
        - *Example* Home page

Dynamic Rexx Server Pages

- Internal processing of .rsp files in Rexx/WS
  - Contents parsed for **<?rexx ...?>** tags
    - Text inside tags are Rexx commands (except for ::include statements)
    - Inserted into memory as is
  - Any ::include statements inside tags:
    - Contents of ::include files inserted into memory after above parsing
    - This is done recursively for maximum 10 levels
  - Text outside of **<?rexx …?>** tags is inserted into memory wrapped in SAY commands
  - The memory is executed instore as Rexx code

Static Rexx Server Pages

- Internal processing of .srsp files in Rexx/WS
    - Check date/time of source file
        - If earlier than corresponding .html file
            - Mark for regeneration and return
        - Read cache file (list of ::included files)
            - If any of these are earlier than corresponding .html file, mark for regeneration and return
    - If regeneration required, process .srsp as .rsp otherwise serve the existing .html file

Client/Server Interface

- !SESSION.
  - Session variables that last for the duration of the client's session
    - eg. !SESSION.MEMBERID saved when logging on
- !GET.
  - Data sent from client to server when issuing an HTTP *GET* command
    - eg. http:/my.com/my.rsp?id=3&num=2
    - !GET.ID = 3, !GET.NUM = 2

Client/Server Interface

- !POST.
  - Data sent from client form to server when issuing an HTTP *POST* command
    - eg. http:/my.com/my.rsp
    - When form contains:
      - username = mark
      - password = fred
    - !POST.USERNAME = mark
    - !POST.PASSWORD = fred
  - Also used for file uploads
- ___RexxLA___ Cookie
  - Used to identify session

Questions

?